Draft
NetDMR Security Specification
Task Assignment 2, Deliverable 5

Prepared for:

Environmental Council of States
444 N. Capitol St. NW
Suite 445
Washington, D.C. 20001

Prepared by:

Eastern Research Group, Inc.
14555 Avion Parkway
Suite 200
Chantilly, VA 20151-1102

December 4, 2007

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

1.0 INTRODUCTION

The Environmental Council of States, the Texas Commission on Environmental Quality, 12 states, EPA's Office of Environmental Information, and EPA's Office of Enforcement and Compliance Assurance are partnering under an EPA Challenge Grant to design, develop, and distribute NetDMR. NetDMR is a web-based application that will allow National Pollutant Discharge Elimination System (NPDES) permittees to submit electronic discharge monitoring reports (eDMRs) to EPA's data system for water permits, the Integrated Compliance Information System (ICIS)-NPDES database. NPDES permits are issued under the authority of the Clean Water Act.

This security specification describes the security processes and standards that NetDMR must use. Many of these processes and standards are also described in the NetDMR Cross Media Electronic Reporting Rule (CROMERR) Checklist and the NetDMR Software Requirement Specification (SRS).

Additional detail for the security specifications described in this document will be provided, as appropriate, in NetDMR Software Design Documents (SDDs). The following SDDs are expected to be prepared for NetDMR:

- Common Components SDD,
- Administrator SDD, and
- Facility User Interface SDD.

1.1 Acronyms

Table 1-1 Acronyms

Acronym	Description and Notes
CDX	Central Data Exchange - http://www.epa.gov/cdx/
COR	Copy of Record, legally enforceable copy of DMR submission
CROMERR	Cross-Media Electronic Reporting and Recordkeeping Rule

Acronym	Description and Notes
DMR	• Discharge Monitoring Report • Required under the Clean Water Act, repots on pollutants or other properties for water discharged into rivers, lakes, streams, etc. (other water bodies)
ECOS	Environmental Council of States
eDMR	Electronic DMR system
ICIS	Integrated Compliance Information System http://www.epa.gov/compliance/data/systems/modernization/index.html ICIS, a Web-based system, enables individuals from states and EPA to access integrated enforcement and compliance and NPDES data from any desktop connected to the Internet. EPA's ability to target the most critical environmental problems will improve as the system integrates data from all media. The public can access some ICIS data through ECHO.
ICIS-NPDES	Integrated Compliance Information System - National Pollutant Discharge Elimination System http://www.epa.gov/compliance/data/systems/index.html
IIS	Internet Information Server
IPT	Integrated Project Team
JAD	Joint Application Design Session
J2EE	Java 2 Platform, Enterprise Edition
NAAS	Network Authentication and Authorization Services
NEIEN	National Environmental Information Exchange Network http://www.epa.gov/exchangenetwork/index.html http://exchangenetwork.net/
NPDES	National Pollutant Discharge Elimination System
OECA	EPA Office of Enforcement and Compliance Assurance http://www.epa.gov/compliance/
OEI	EPA Office of Environmental Information
SAK	Secure Authentication Key
SDD	Software Design Document
SRS	Software Requirements Specification
SSL	Secure Socket Layer
TCEQ	Texas Commission on Environmental Quality
URL	Uniform Resource Locator
XML	eXtensible Markup Language
XSLT	XML style sheet

1.2 References

- 830-1993: IEEE Recommended Practice for Software Requirements Specifications

- TCEQ Software Requirement Specification Template (http://www.dir.state.tx.us/pubs/framework/gate2/sdlc/index.htm)

- Cross-Media Electronic Reporting Rule, 40 CFR Part 3

- Federal Information Processing Standards (FIPS)-approved algorithms for generating Message Digest (http://www.csrc.nist.gov/CryptoToolkit/tkhash.html)

- FIPS-approved algorithms for generating/verifying digital signatures (http://www.csrc.nist.gov/CryptoToolkit/tkdigsigs.html)

- NIST Hash Function Policy (http://www.csrc.nist.gov/pki/HashWorkshop/NIST%20Statement/NIST_Policy_on_HashFunctions.htm)

- SecureRandom Specification (http://java.sun.com/j2se/1.4.2/docs/api/java/security/SecureRandom.html)

- Security Salts (http://msdn.microsoft.com/msdnmag/issues/03/08/SecurityBriefs/)

- Spring Security module (http://www.acegisecurity.org/)

2.0 ACCESSING NETDMR

This section defines how a user accesses NetDMR. All NetDMR web pages will be classified as protected or publicly available. Protected pages will be accessible only over Secure Sockets Layer (SSL) protocol v3 or Transport Layer Security v1.0 [REQ 125] and will require that the user provide the user name and password associated with their account for verification by NetDMR prior to allowing accessing. Section 3.0 specifies how a user creates an account. Protected pages are further classified to limit which authenticated users can access the page [REQ 109]. Section X.X describes Access Control and applicability to users and pages. Public pages will also be accessible over SSL, but will not require a user to provide identity information. Anyone can browse to the NetDMR web site and view public pages.

NetDMR requires that users access the web site using one of the following supported internet browsers [REQ 263]

- Internet Explorer 6.x
- Internet Explorer 7.x
- Mozilla Firefox 2.x

NetDMR functionality and performance will not be tested or validated using other browsers. Users must also have JavaScript enabled in their browser [REQ 265] to access NetDMR functionality.

3.0 ACCOUNT CREATION

This section outlines the NetDMR account creation process and describes the technical aspects of the security measures that control the process (e.g., generating a secure random number). The Common Components SDD will provide additional implementation details of the registration process, as well as screen captures of the web interface prototype pages.

There are two account types that can be created in NetDMR, Installation and Instance accounts. Installation accounts are global to the NetDMR installation and are linked to a specific NetDMR instance. Instance accounts are specific to an instance within an installation. Each account type is associated with different user types. A user type identifies a profile of a type of user that will be accessing NetDMR. Each user type is assigned a set of roles and permissions that users of that type can be granted. When a NetDMR account is created, both the account type and user type must be specified. For more information on the different types of users see Section 7-4. The process used to create the user account depends upon the type of account that is being created.

3.1 Installation Account Type

As described in Section 7-4, the System user type is the only user type that is associated with a NetDMR installation account. The System Administrator role is the only role that can be assigned to a System User. A System Administrator creates instances within a NetDMR installation. A NetDMR installation, without instances, can not be used to submit eDMRs.

NetDMR will provide a process to create one or more System Users that is separate from the process used to create instance accounts. It is expected that only a few System Administrators will be created for an installation. The process to create the initial System Administrator will be run immediately after deploying NetDMR, but can be used anytime thereafter if another System Administrator account is required. To protect against unauthorized account creation, the NetDMR configuration file contains an initialization flag and an initialization key. The flag indicates whether the initialization process can be run. The key must be provided in the initialization process. The initialization process includes the following steps.

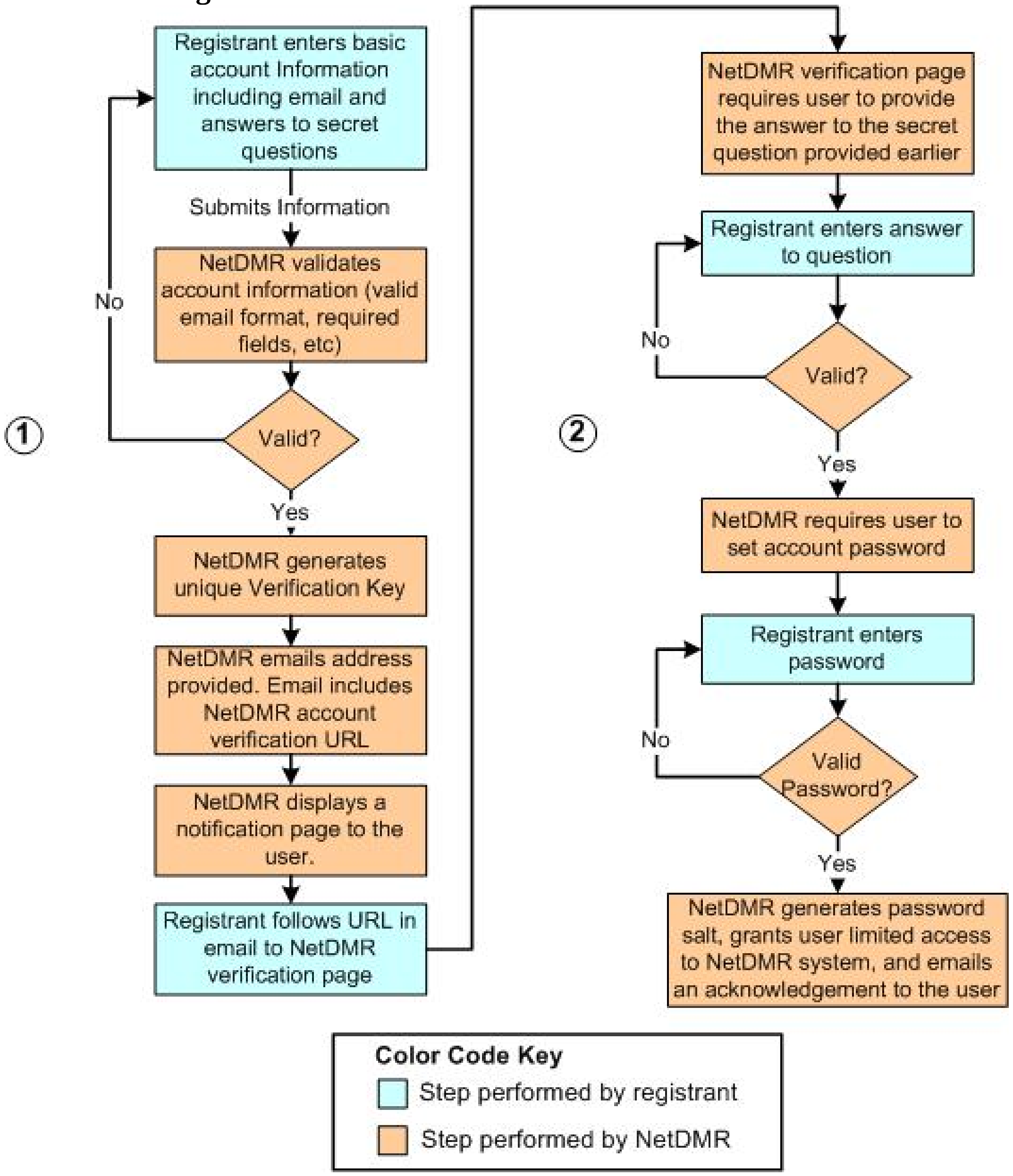
1. User deploys NetDMR:
    a. Set initialization flag to allow initialization.
    b. Set initialization key to some string.
2. User accesses initialization page:
    a. Enter initialization key.
    b. Enter System Admin user account information.
3. If key is correct, user verifies the information provided.
4. NetDMR creates the specified account.
5. NetDMR displays a confirmation page.
6. User sets the initialization complete flag to prevent initialization.

After these accounts have been created, the initialization complete flag in the NetDMR configuration file should be set to prevent access to the initialization process. If additional System Administrators are required at a later date, the flag can be temporarily reset.

3.2 Instance Account

A System Administrator will create one or more instances within NetDMR. Once an instance is created instance accounts can be created for the instance. Figure 3-1 provides the process used to create instance accounts.

Figure 3-1. NetDMR Account Creation Process



1
Registrant enters basic account Information including email and answers to secret questions
Submits Information
NetDMR validates account information (valid email format, required fields, etc)
No
Valid?
Yes
NetDMR generates unique Verification Key
NetDMR emails address provided. Email includes NetDMR account verification URL
NetDMR displays a notification page to the user.
Registrant follows URL in email to NetDMR verification page
2
NetDMR verification page requires user to provide the answer to the secret question provided earlier
Registrant enters answer to question
No
Valid?
Yes
NetDMR requires user to set account password
Registrant enters password
No
Valid Password?
Yes
NetDMR generates password salt, grants user limited access to NetDMR system, and emails an acknowledgement to the user
Color Code Key
Step performed by registrant
Step performed by NetDMR

The Common Components SDD will describe the account creation process in detail. The security assurances associated with this process include:

- Verification that the user has access to the email address he/she provided during registration [REQ 80].
- Verification that the user validating the email address and choosing a password is the same user that completed the account creation process [REQ 83.]

3.3 Verify User Access to Email Address

Verification of the user's access to the email provided during the account creation process is performed by sending a message to that email account. The user must follow a URL provided in the email to complete the account creation process and receive a NetDMR account. The verification key included as part of the URL makes it extremely difficult for a malicious user to determine a valid URL without access to the email.

The verification key will be automatically prepared by NetDMR using an algorithm that generates a random unique key. The key will be constructed as follows:

1. Generate 128 random bytes (1024 bits) using the approved SHA1PRNG random number generator.
2. Concatenate (1) with the username, IP of the requesting user, and the current system time
3. Generate a hash of the resultant string using the approved SHA256 hashing algorithm
4. Convert the output of the hash into a hexadecimal string

The output of Step 4 is a 64 length character string of hexadecimal characters. For example:

```
8bc28ed4bfaa2d74b3d16f16f0d3ffcba609a500f96484c3a24c25d92e965dac
```

The URL included in the email must include the hexadecimal string as a query parameter. For example:

https://netdmr.example.com/verifyAccount.web?verificationKey=*8bc28ed4bfaa2d74b3d16f16f0d3ffcba609a500f96484c3a24c25d92e965dac*

NetDMR will validate that the verificationKey parameter was produced by NetDMR and determine the user account for which the key was generated.

3.4 Verify the User Completing the Account Creation Process

It is possible that a malicious user may gain access to the email sent to the user attempting to create a NetDMR account. For example, the registrant may have entered the incorrect email address, the registrant may leave the email open on his/her desktop and step away from the office, or the malicious user may intercept the email in transit. To mitigate the risk that a malicious user can masquerade as the registrant by completing the registration process the following measures must be taken:

- After following the verification URL, the registrant will be presented with a page that requires the user to respond correctly to one of the security questions he/she answered during the initial registration process and to provide an acceptable password. [REQ 83]
- If the registrant provides an incorrect response to a security question three times, the verification key becomes invalid and the user is sent a notification email. The user must contact the Regulatory Authority to generate a new verification key and complete the account creation process. [REQ 83]
- The verification key will only be valid for only 60 days. After 60 days, the user must re-start the account creation process or contact the Regulatory Authority to generate a new verification key. [REQ 83]
- A user attempting to use an invalid verification key will be shown a message stating the reason the key is no longer valid. [REQ 87]
- A message will be sent to the email address supplied during the registration process after the verification process is completed. [REQ 88]

4.0 USER ACCOUNTS

This section describes measures that will be implemented to secure user accounts and account information, including the use of a salt to protect information in the database, composition requirements for passwords, and detailed information on the use of security questions.

4.1 User Account Salt

NetDMR requires the creation of a unique 8 character random salt for each user. A salt is a set of characters that is appended to plain text prior to creating a hashed value of the plain text. For more information on salts see http://msdn.microsoft.com/msdnmag/issues/03/08/SecurityBriefs/. Salts are commonly used to strengthen the protection of user passwords as follows:

- The addition of the user-specific salt to each user's password assures the salt+password combination for each user is unique. A one-way hashing algorithm assures that the hashed forms of any two distinct values do not hash to the same value (defined as a collision). While such collisions do occur, the likelihood of such collisions is remote. Appending a user-specific salt to the password prior to hashing significantly decreases the likelihood that two users with the same password will have the same hashed password.
- Makes it extremely difficult for a malicious user to apply a pre-generated list of the hashed version of common passwords to determine a user's password. A malicious user would also need the user's salt value to create a pre-generated list of hashed passwords for each user.

The salt must be created using the approved random number generator listed in Section 10. While it is unlikely that the random number generator will generate the same random salt for multiple users, NetDMR will verify each generated salt is unique in the database prior to assigning it to a user.

NetDMR will use the user-specific salt to protect the password and answers to all security questions for the user account. See Section 3.3 and 3.4 for more information on application of the salt in these situations. The salt will be created during the initial registration process and stored with the user account. The salt value will be stored as plain text within the NetDMR identity store database tables

4.2 User Name

The user name uniquely identifies a NetDMR user account. Two active accounts can not have the same user name. Each user name must meet the following requirements:

- User name can be up to 50 characters in length
- A user name can be the same as the user's email address [REQ 111]
- A user name can be changed at any time

Allowing use of the email address provides additional flexibility for setting the user name, and may make it easier for users to remember the user name specified for the account. However, if the user chooses this option, and subsequently changes the email address associated with the account, the username for the account should be automatically updated to reflect the new email address.

4.3 Email Address

Each active user account must be associated with a unique email address within a NetDMR instance. The email address is initially used when creating an instance account to verify that the user has access to the specified email account. The email address is subsequently used to send various notifications (e.g., submission acknowledgements) and to allow a user to reset a forgotten password.

4.4 Password

NetDMR uses a password to authenticate that a user is the owner of the account. While it is impossible to prevent users from choosing a weak password (e.g., the user's birthday, dog's name, etc.), NetDMR must apply some business rules to strengthen user password selection. User passwords must meet the following requirements:

- Must be between 8 and 20 characters in length and contain both letters and numbers [REQ 110].
- Must be case sensitive.
- Must be comprised of only the following characters:
  - Uppercase letters (ABCDEFGHIJKLMNOPQRSTUVWXYZ)
  - Lowercase letters (abcdefghijklmnopqrstuvwxyz)
  - Numbers (0123456789)
  - Special characters (!@#$^&*+=)
- Must be changed at least once every 12 months* [REQ 112].
- Users must be required to enter the new password twice when changing their password.
- Must not be the same as any of the last 10 passwords used for the account.
- Can be changed by user at any time [REQ 112].
- Must be stored in the database in a hashed format [REQ 113].
- Must be appended with a unique 8-character password salt prior to storing in the database [REQ 114].
- Must not be the same as the answer to a security question

Each NetDMR Regulatory Authority can require users to change the account password more frequently than every twelve months. For example, EPA Regions will require users to change account passwords every 90 days.

The following steps must be followed when storing a user's password in the NetDMR identity store:

1. User provides the plain text password.
2. NetDMR retrieves the password salt associated with the user's account.
3. NetDMR appends the salt to the user-provided password.
4. NetDMR creates a hash of the string generated in Step 3 using the approved SHA-256 algorithm.
5. NetDMR converts the output of the hash in to a hexadecimal string that is 64 characters long.
6. NetDMR stores the hexadecimal string in the NetDMR identity store as the user password.

Conversion to a hexadecimal string is performed to avoid possible character encoding issues in the database. When validating that a provided password corresponds to the account password (e.g., during log in) a similar process will be followed, except that the string generated in Step 5 will be compared to that stored for the account. If the strings are identical, the user provided the correct account password.

Users frequently forget the password associated with their account. The process for resetting a password in NetDMR is similar to the process for setting the initial password for the account. The user accesses password reset functionality and provides the answer to a security question. After providing a correct response to the security question, NetDMR generates a verification key and sends a message to the user’s email address. The user completes the process of resetting his/her password by following the steps outlined in Section 3.2 for creating the initial password.

4.5 **Security Questions**

NetDMR will use security questions to provide additional application security.. A security question is a simple question on a topic such as ‘What was the name of the high school you attended,’ with which the user is very familiar. NetDMR requires all users to answer at least one security question during the registration process [REQ 82]. However, each Regulatory Authority associated with a NetDMR installation can customize the number of security questions its users must answer during the registration process [REQ 296]. Users will choose their security questions from a static list of ten questions provided by NetDMR [REQ 297]. The security questions the user answers are registered for that user account. Answers to security question must meet the following requirements:

- Case sensitive
- Can only include the following characters:
  - Uppercase letters (ABCDEFGHIJKLMNOPQRSTUVWXYZ)
  - Lowercase letters (abcdefghijklmnopqrstuvwxyz)
  - Numbers (0123456789)
- Can be changed by the user at any time.
- Must be stored in the database in a hashed format.
- Must be appended with a unique 8 character salt prior to storing in the database.
- Must not be the same as the account password.
- Must not be the same as the answer to one of the other security questions.

If an account has multiple registered security questions, one of the questions will be randomly chosen any time a security question is required. There are three primary applications of security questions by NetDMR:

- In conjunction with the account password. For this application, the user is required to enter both the account password and a response to a security question. Failure to provide the correct information for either will result in an error. This use case provides additional security beyond entering the password (e.g., signing a DMR, changing account information).

- In conjunction with an email. For this application, the user receives an email after responding to a security question. The email either contains the requested information or provides additional steps to perform the requested action. The account password provides stronger protection than the security question. This use case is used when the password to the account is not known. See Section 4.3 for more information on use of this process to reset an account password.

- By itself. For this application, the requested action is performed after a correct response to a security question is provided. This use cases is appropriate only for actions that require minimal security and do not reveal confidential information. For example, this process will be used to display the user name for an account if the email address is known.

Users must be able to change the security questions and/or answers registered to their account at any time. Internal Administrators must be able to provide a temporary answer to a security question associated with a user account to assist a user that forgets the answers to the security questions on his/her account. When an administrator sets a temporary answer, it removes any and all security questions previously registered to the account. The administrator must contact the user to provide the temporary answer. Following the next authentication, the user is required to register the appropriate number of security questions for his or her account.

The process for storing the answers to security questions in the NetDMR identity store is exactly the same process as that used to store passwords. See 4.3 for more information.

5.0 LOCKED ACCOUNTS

NetDMR Accounts can be locked for numerous reasons, including:

- After three consecutive unsuccessful login attempts within a 24 hour period [REQ 118].
- After three consecutive unsuccessful attempts to sign a DMR due to providing an invalid password or response to a security question.
- After three consecutive unsuccessful attempts to change account information due to providing an invalid password or response to a security question.
- Internal Administrator locks account due to suspected compromise.
- User locks account due to suspected compromise.
- Permit Administrator locks account due to suspected compromise.

If the account is locked as a result of invalid login attempts, DMR signing attempts, or account information change attempts, the user can unlock his/her account by providing a valid response to a security question. After answering the security question, an email that includes a verification URL will be sent to the user, similar to the process outlined in Section 3.2 when initially creating the account. The user is required to follow the link in the email, and set a new password on the account. After these steps are completed, the account will be unlocked.

If the account was locked by the user, an Internal Administrator, or Permit Administrator, the account can only be unlocked by an Internal Administrator.

6.0 USER AUTHENTICATION

User authentication is performed when a user attempts to log in to NetDMR. The user must provide the valid user name and password for the account. NetDMR will compare the provided password for the specified account as described in Section 4.3.

After successful authentication, the user is considered authenticated for the lifetime of the web session. The session is invalidated after a period of 30 minutes of inactivity [REQ 226]. After a session is invalidated, the user must login to NetDMR again. Invalidated sessions lose all information and state contained within the session. If the user was in the middle of completing a DMR when the session was invalidated (e.g., user left their computer for 40 minutes), the information the user entered will be lost and will have to be re-entered. Users can save partially completed DMRs [REQ 171] at any time.

Certain NetDMR actions require the user to re-authenticate with the account password and a response to a security question (e.g., signing DMR, changing user information). This protects against a malicious user masquerading as the logged in user if the user temporarily leaves his/her computer unattended.

Following three unsuccessful authentication or re-authentication attempts, the user account is locked (see Section 5.0); the user can not continue using the account until it has been unlocked. If the account was locked due to failed re-authentication attempts, the state of the session must not be lost until after 30 minutes of inactivity. This provides the user with the opportunity to unlock the account without losing the current session state.

7.0 ACCESS CONTROL

This section specifies the processes that will be used to determine whether a user can perform a specified action in NetDMR. NetDMR access control is defined through three aspects:

- Permissions,
- Roles, and
- User Types.

Permissions directly relate to a specific action (e.g., SignDMR) and are the finest granularity used for access control. Permissions are grouped into roles to simplify management and assignment to users. User Types describe overall classes of users. Certain roles may be applicable to only certain User Types These three topics are described in more detail in the following sections.

This section describes the overall framework for NetDMR access control, each SDD will describe the application of the framework in detail, as appropriate. For example, each SDD will specify the permissions that are applicable to the functionality included in the SDD, as well as the roles with which those permissions are associated.

7.1 Permissions

Permissions define the foundation for the NetDMR access control. Each action that a user can perform using NetDMR is associated with a permission. For example, searching CORs could be associated with a *SearchCOR* permission. NetDMR will make access decisions based on whether the user has the appropriate permission for the requested action. Each permission is classified into one of the four categories, depending on the context of the permission. Table 7-1 lists the categories and provides example permissions that could apply to that category.

Table 7-1 Permission Categories

Permission Category	Description	Example Permissions
DMR	Actions that can be performed on a DMR.	EditDMR SignDMR ViewPartialDMR
Permit	Actions that can be performed on a Permit	SearchCORs ManagePermitAdminRole
Instance	Actions that are specific to a NetDMR instance	LogIn
System	Actions that are specific to a NetDMR installation	CreateInstance

7.2 Roles

To simplify assignment of permissions to groups of users, permissions are assigned to roles and roles are assigned to users. This allows for quick and easy modification of roles associated with groups of users. For example, all users with a certain role will be affected immediately if the permissions associated with the role are modified. Roles are classified in one of four categories based on the categories of permissions it can contain. Table 7-2 lists the role categories.

Table 7-2 Role Categories

Role Category	Permission Categories	Example Roles
DMR	DMR	ViewPartialDMR
Permit	DMR Permit	PermitAdmin Viewer Editor Signatory
Instance	DMR Permit Instance	Access InternalUser InternalAdmin
System	System	SystemAdmin

When a DMR Role is assigned to a user, the DMR for which the role is assigned must also be included. The user can only perform the associated DMR permission actions in the role for the specified DMR.

When a Permit Role is assigned to a user, the permit ID for which the role is assigned must also be included. The user can only perform the associated Permit permission actions for the specified permit. In addition, the user can only perform the associated DMR permission actions for the DMRs associated with the specified permit.

When an Instance Role is assigned to a user, the instance for which the role is assigned must also be included. The user can only perform the associated instance permission actions for the specified instance. The user can perform the Permit permission actions for all permits associated with the instance. The user can also perform the DMR permission actions for all DMRs associated with the instance.

System roles can contain only System permissions.

7.3 User Types

In addition to roles and permissions, NetDMR supports User Types. User Types are mutually exclusive classes of users; a user account can only be associated with only one user type. The roles that can be assigned to a user vary depending on the user type. As described in Section 3.0, NetDMR has two account types: installation accounts and instance accounts. Each user type is associated with one of these account types. Table 7-3 lists the user types and the roles that can be assigned to each.

Table 7-3 User Types

User Type	Account Type	Description	Role Categories	Example Roles
Permittee	Instance	Permittee staff are responsible for completing DMRs.	DMR Permit Instance	Access Signatory Viewer Editor PermitAdmin
Data Provider	Instance	Data Providers may enter the DMR data for a permittee, but are not allowed to sign the DMR.	DMR Permit Instance	Access Viewer Editor
Internal	Instance	Internal users are Regulatory Authority	DMR Permit	Access InternalUser

User Type	Account Type	Description	Role Categories	Example Roles
		staff	Instance System	InternalAdmin SystemAdmin
System	Installation	System users are responsible for maintaining the installation of NetDMR.	System	Access SystemAdmin

With the exception of the System User Type, users specify the User Type for their account during the account creation process. System users are created through an internal process rather than using the web-based NetDMR account creation process. System users can only be granted System roles. The process for creating the different types of users will be described in more detail in the Common Components SDD.

Each NetDMR role must specify the User Types with which it can be associated. For example, as shown in Table 7-3, all User Types can be assigned the Access role, while only Internal User Types can be assigned the InternalUser role. NetDMR will include processes that enforce assigning of roles to appropriate User Types.

8.0 USER AUTHORIZATION

As described in Section 7.0, NetDMR access control relies on the permissions granted to a user. Each permission is classified as a DMR, Permit, Instance, or System permission, based on the context in which the permission will be applied. For example, a DMR permission refers to a permission that takes place in the context of a specific DMR (e.g., editing a DMR). NetDMR is provided the permission that is being checked, along with the appropriate context for checking the permission, to complete access control.

For example, to verify whether a user has permission to search CORs for a specific permit (a Permit Permission) NetDMR would be provided the name of the permission, *SearchCOR*, and the context, the Permit ID. In this example, NetDMR would follow the steps below:

1. Determine which instance roles contain the permission.
2. If the user has been assigned one of the instance roles from Step 1, the user has the appropriate permission.
3. If the user has not been assigned one of the instance roles, determine which permit roles contain the appropriate permission.
4. Determine if the user has been assigned one of the permit roles from Step 3 for the particular permit.
5. If the user has been assigned one of the permit roles, the user has the appropriate permission.

In some cases it may only need to be determined if the user has the *SearchCOR* permission on any permit rather than for a specific permit. This type of check will also be supported.

9.0 IDENTITY AND POLICY MANAGEMENT

Identity and policy management encompass a broad spectrum of functionality related to storing and using identify and policy information.

Identity and policy management rely on the application identity and policy store. An identity store is a location that houses user account information. A policy store relates a user account to a set of permissions that specify the actions that a user can perform or the information that a user can access. Account and policy information can be stored and accessed in numerous ways, including using flat files, a database, or a Lightweight Directory Access Protocol (LDAP) server.

Identity and policy stores comprise only a small portion of the overall solution for identity and policy management. Numerous business processes must also be implemented to manage the NetDMR stores, including:

- Limiting the account information that users can change.
- Password change frequency.
- Protecting the password within the store from unauthorized access.
- Locking/Unlocking accounts.
- Automatic emails (e.g., account creation or modification).
- Limiting users to a single user session.
- Logging specific events.
- Displaying logs.
- Fraud analysis.
- Displaying past logins (e.g., last 10 logins).
- Retrieving username.
- Resetting password.

Implementation of these business processes will provide NetDMR a robust security solution that meets the NetDMR requirements. When evaluating identity and policy store solutions for NetDMR, both the method used to store user and policy information as well as whether the solution can be used to meet all the associated business processes were considered. When considering Single Sign On solutions, the other applications that use the solution must also be evaluated to assure that the security required for NetDMR is not compromised by lower security requirements in the other applications. For example, NetDMR requires a password length of between 8 and 20 characters and provides functionality to allow users to set the password through a NetDMR interface. If accounts are shared with another application that allows users to set a password with a length between 4 and 7 characters, the NetDMR password length rule could be easily circumvented.

After considerable discussion with stakeholders it was decided that the NetDMR application will include default identity and policy stores implemented within the NetDMR database. This allows the greatest flexibility for implementing the various business processes that wrap the stores, while keeping the deployment and maintenance of the application simple. The use of LDAP is a common solution for storing this information as well. However, in many cases, an LDAP solution is deployed as part of an enterprise wide solution where the policies and procedures surrounding its use will vary by enterprise.

Although NetDMR will store identity and policy information in the database, the NetDMR design will not preclude the use of different solutions, such as LDAP. NetDMR will use the standard open source Spring Security Framework to perform authentication and authorization. This Framework allows different methods of performing authentication and authorization to be substituted for the default implementation. See Section 13 for more information on Spring Security.

10.0 NETDMR DATABASE

NetDMR must have access to a relational database to store DMRs, account information, logs, and other information. NetDMR will use two accounts to create, read, update, and delete (CRUD) the information in the NetDMR database tables.

The first account will be used to CRUD all records for all NetDMR tables except the log tables. The account will have create and read permissions for records in the log tables. A second database account will have read and delete permissions for records in the log table, but not insert or update permissions. The segregation of log permissions between the accounts allows an independent process to be used to purge the application logs.

The protocol used for the communication between the NetDMR application server and the NetDMR database server is specific to the environment in which the application is deployed. Various database platforms support different protocols, agencies have different standards for how the connections should be managed (e.g., through an Oracle Connection Manager), and the topography of the server environment may vary within each environment.

The database connection information, including the database accounts, will be stored in a NetDMR configuration file.

11.0 NetDMR and Exchange Network

NetDMR must have a Network Authentication and Authorization Services (NAAS) account to send and retrieve information over the Exchange Network. The account must have the appropriate permissions on the application Exchange Network Node to call the required services. The list of services will be defined through the Permit, ICIS-NPDES Batch, and Error Message Integrated Project Teams. A Secure Authentication Key (SAK) must be created for this account. See http://www.exchangenetwork.net/node/dev_toolbox/sak.htm for more information on SAKs. The account information will be stored within a NetDMR configuration file.

All communication over the Exchange Network will occur over SSL.

12.0 SECURITY STANDARDS

This section outlines the security standards, specifications, and algorithms that will be used in NetDMR.

Table 12-1 NetDMR Security Standards

Type	Standard	Description
Secure Sockets Layer (SSL) specification	SSL v3 or TLS v1 [REQ 125]	Encrypts the communication between the client browser and the NetDMR web server.
Digital Certificate	x.509 Certificate with RSA 1024 bit keys [REQ 122]	The private key accompanying this certificate will be used for signing NetDMR CORs. The certificate can be used to verify the signature.
Hash Algorithm	SHA-256 [REQ 121]	An algorithm that turns a variable-sized amount of text into a fixed-sized output (hash value). NetDMR uses hashing to protect original text from discovery (e.g., passwords, secret answers) and to generate a reproducible “fingerprint” to verify information has not been changed (e.g., within Copy of Record).
Signature Algorithm	SHA256withRSA	SHA256withRSA specifies that the plaintext should be hashed using the SHA256 algorithm, and then sign the hash with the RSA algorithm.
Random Number Algorithm	SHA1PRNG	The SHA1PRNG is an algorithm for generating pseudo random numbers. This will be used by NetDMR when a random number is required (e.g., verification keys, password salts).

The SSL configuration depends on the environment in which NetDMR is deployed (i.e., web server). Appropriate SSL certificates can be purchased from vendors such as Verisign (http://www.verisign.com/) and Digicert (http://www.digicert.com/). Configuration of the certificate in the deployment environment is outside the scope of this document.

Similar to SSL certificates, digital certificates used for signing documents can be provided by numerous vendors. The Exchange Network has an established infrastructure to generate certificates that can be used for this purpose. It is anticipated that the Exchange Network will provide the appropriate certificate for NetDMR.

Implementations of the specified hash, signature, and random number algorithms are available from vendors such as Sun, IBM, and The Legion of the Bouncy Castle. NetDMR can be configured to use any of these implementations, provided they implement the specified algorithm, without changing any NetDMR code. This is accomplished by registering the implementation, referred to as a *Provider*, through the use of Java Cryptography Extensions (JCE). More information about this process can be found at http://java.sun.com/j2se/1.5.0/docs/guide/security/jce/JCERefGuide.html. The Sun Provider is registered by default in the Sun JVM, and will be the default Provider used by NetDMR. When applicable, each SDD will provide additional detail on when and how the required algorithms will be used. For example, the Common Component SDD will describe the hashing of passwords, the Administrator SDD will describe storing the digital certificate used for signing Copy of Records (CORs), and the Facility User Interface SDD will describe the process used to sign CORs.

13.0 SECURITY FRAMEWORK

Numerous security frameworks have been developed to support authentication and authorization management. NetDMR will use the Spring Security module (http://www.acegisecurity.org/), formerly known as Acegi Security. Spring Security is a powerful, flexible open source security solution for enterprise software, with a particular emphasis on applications that use Spring. Spring Security provides applications with comprehensive authentication, authorization, instance-based access control, channel security and human user detection capabilities. The framework provides numerous hooks and default implementations for authentication and authorization decisions against different types of identity and policy stores such as a relational database, LDAP, and Central Authentication Service (CAS). Through the use of the Spring Security Framework, NetDMR can be migrated to a different identity and policy store with minimal impact on existing code.